Fault Injection method for Safety and Controllability Evaluation of Automated Driving

Garazi Juez Uriagereka[1] and Ray Lattarulo[1] and Joshue Pérez Rastelli[1]
and Estibaliz Amparan Calonge[1] and Alejandra Ruiz Lopez [1] and Huascar Espinoza Ortiz [1]

Abstract— **Advanced Driver Assistance Systems (ADAS) and automated vehicle applications based on embedded sensors have become a reality today. As road vehicles increase its autonomy and the driver shares his role in the control loop, novel challenges on their dependability assessment arise. One key issue is that the notion of controllability becomes more complex when validating the robustness of the automated vehicle in the presence of faults. This paper presents a simulation-based fault injection approach aimed at finding acceptable controllability properties for the model-based design of control systems. We focus on determining the best fault models inserting exceptional conditions to accelerate the identification of specific areas for testing. In our work we performed fault injection method to find the most appropriate safety concepts, controllability properties and fault handling strategies at early design phases of lateral control functions based on the error in the Differential GPS signal.**

I. Introduction

Automated vehicle technology has the potential to redefine the automotive world and will definitely bring major benefits for road safety, emissions and congestions. With the growth of control complexity and the reduction of the drivers role, many challenges arise with respect to the safety and controllability risk assessment of these vehicles. Therefore, additional focus needs to be given to smart safety concepts [1], such as the accounting of stringent new conditions when performing hazard analysis. The most critical vehicle functions demand fail-operational behavior, as the system cannot simply shut down silently, i.e. fail-silent behavior is not acceptable for highly automated driving. Thus to reach the highest safety- critical levels, such systems should work on a fail-operational manner achieved by either redundancy or alternative functions. Furthermore, traditional validation and verification methods might not be sufficient, especially to perform combinations of exceptions in unusual operation conditions. A promising approach to overcome this limitation is Fault Injection (FI) [2][3]. Svenningson [4] investigated how to benefit from conducting FI experiments on behavioral models of software. This approach is defined as model-implemented FI , since a model is extended with artefacts to support the injection of fault effects during simulation. In particular, it addressed injection of hardware fault effects into Simulink models. Another similar approach is introduced in [5]. The FISCADE FI tool is developed as a plug-in to SCADE (Safety-Critical Application Development Environment) and it automatically replaces original operators with FI nodes. However, the fault effects are not considered at level of vehicle dynamics. This is of particular interest when calculating critical parameters such as the Fault Tolerant Time Interval (FTTI), which is directly related to the controllability of the vehicle. As described in [6], FTTI is the time the system has to transition to a safe state after a failure has occurred and if safe state is not reached within this interval, an emergency operation shall be specified.

In [7], Silveira introduced a Matlab/Simulink-based co-simulation framework for evaluating the stability of electrical vehicles using fault injection. This latter work did not analysed controllability challenges and fail-operational issues, which are relevant factors for automated driving.

The work underlying this paper intends to develop a simulation-based fault injection framework to: (i) get testing data regarding failure modes and failure effects of automated critical functions as a way to complement standard safety analysis techniques, (ii) calculate the FTTI which is directly related to the controllability of vehicles, (iii) evaluate and improve the robustness of automated functions, and (iv) obtain trade-off evaluation results between safety and cost issues, already at concept level. Our approach has been evaluated in a use case of a lateral control function for an urban vehicle.

The reminder of this paper is structured as follows. In Section II, we present the relevant background w.r.t FI in automated driving. Thereafter, Section III describes our simulation-based FI approach. Afterwards, an use case targeting lateral control is explained in Section IV. Finally, Section V presents conclusions leading to an outlook on future work.

II. Fault injection in Automated driving

Among the unique challenges of designing automated cars is ensuring the ability to avoid a specified harm or damage through the timely reactions of the vehicle, assuming the driver is out of the loop. We refer to this ability as controllability [8]. ISO 26262 introduced a similar controllability definition, but centered on the drivers ability to control the vehicle. The next version of ISO 26262 will need to adjust the controllability definition to highly automated driving. Our work pursues the testing technologies that can predict: (1) the acceptable controllability properties (such as computation delays) for a given electronic architecture, (2) what additional design areas related to dependability assurance must be improved or added, and (3) whether we need to concentrate more testing in specific areas to guarantee the robustness of the vehicle against harms. In particular, controllability is directly related to the time- span in which a fault or faults can

[1] Tecnalia Research and Innovation, Derio, Vizcaya, Spain, 48160. (e-mail: {(garazi.juez,rayalejandro.lattarulo, joshue.perez, Alejandra.Ruiz, estibaliz.amparan, Huascar.Espinoza}@tecnalia.com)

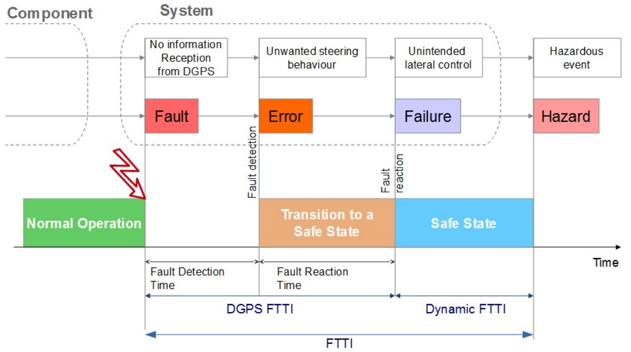


Fig. 1. Fault-Error-Failure Chain and FTTI definition

be present in a system before a hazardous event occurs. This parameter is referred as FTTI and can be better understood with an example for the Fault-Error-Failure chain of the lateral control due to a Differential GPS (DGPS) component failure, as shown in Fig.1. This parameter is crucial when calculating the maximum time for system reconfiguration (instead of a simply shut-down) before a hazardous event occurs.

A. Primer on Fault Injection

In order to better understand the role of FI on safety assessment, a theoretical background on this field is essential. This technique either evaluates or validates the dependability of systems [9]. Dependability is defined as that property of a computer system such that reliance can justifiably be placed on the service it delivers [10]. By exploiting such a testing technique, controlled experiments are conducted by the deliberate injection of faults into the system and the reaction is observed. Its main objectives are to: (i) understand the systems behavior under the effects of real faults, (ii) evaluate the system fault tolerance, (iii) forecast the faulty behavior of the target system, (iv) identify weak links on the design, and (v) estimate the coverage and latency of Fault Tolerance Mechanisms (FTM). Actually, as they are not triggered under normal conditions, FI is used to activate those exceptional conditions and to remove FTM design faults. A detailed description of the different FI techniques and tools is presented in [11]. One of the techniques with more relevant benefits is the so-called simulation or model-based FI which allows full observability and controllability. To get meaningful and accurate FI experiment results, a representative fault model is required. Different types of faults can appear depending on its nature during the system design process or during its operational life [11].

B. Fault Injection requirements in ISO 26262

ISO 26262 recommends the use of FI in different phases of the V model, including both sides of the V-cycle. The main aim of FI on the left side is to check that behavioral specifications do not contain any error or omission in the presence of faults. In general, FI helps to ensure that the system implements the appropriate safety mechanisms that prevent the violation of safety properties [12]. The right side of the V-cycle stresses the verification and validation of safety mechanisms. FI is mentioned at system, hardware (HW)and software (SW) level. At system level, faults are injected into the item by reproducing the possible item malfunctions. This is done to evaluate different safety concepts and safety mechanisms, since these last ones are not invoked during normal operation of the system. This is one of the main emphases of our work. We aim at evaluating different safety concepts based on specific FTTI controllability parameters by simulating item malfunctions in the presence of representative fault models.

III. Simulation-Based Fault Injection Approach

A. Generic Framework

As previously pointed out, one of the main aims of our sim- ulation environment extension is to evaluate properties such as controllability and to trade-off between system dependability attributes and cost already at concept level. To do so, those parameters are evaluated via a simulated vehicle. The Dynacar platform [13] is a real-time vehicle dynamics simulation SW solution based on multiple domains vehicle models. It provides a high-fidelity vehicle physics simulation basing a multibody dynamics models (i.e.: engine, transmission, steering system, braking system, aerodynamics). It permits a real-time simulations, either HW or SW functionalities, combined with its notable modularity and interfacing options. Hence it allows the mixing of virtual or real Electronic Control Units (ECUs), vehicle sensors and vehicle control variables. The FTTI of an item must remain within the limits given by physical properties of the respective functionality e.g., the maximal time span the lateral control is allowed not to be under control without losing vehicle controllability.

The system under test is developed as part of a model-based design control function development. Due to the benefits of using this method, an early verification and validation of the developed automated critical vehicle function can be achieved. In fact, model-implemented FI technique is used where faults are introduced via model blocks i.e. saboteurs and can be inserted into either SW or system models. This allows injecting different errors such as timing, control flow or data by extending behavioral models with FI blocks called saboteurs. It is worth noting that the proposed solution is independent of the selected commercial model-based environment.

On the basis of the so-called FARM (Fault, Activation, Readouts, Measures) model [14], a simulation-based FI framework is proposed. FARM methodology emerges as an effective way to characterize such an environment and follows the subsequent process: a fault characterized by a model, a location, an injection time and a duration is injected into the system. Depending on the executed workload, fault activation trajectories might differ, i.e. activation trajectories specifies how the system is functionally exercised during the experiment. Another important significant variable to define is where to observe systems behavior under fault. This is the main objective of Readouts parameter. Once

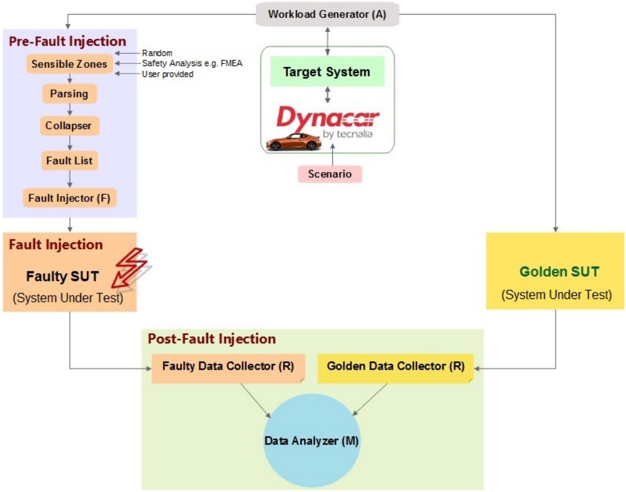


Fig. 2. Simulation-Based Fault Injection Framework: Generic Approach

those results are logged, Measures are calculated so the final dependability of the system can be evaluated.

In addition to the FARM model, another well-established FI concept is set up as basis of our approach. Bearing in mind that the selected method is based on a simulation environment, a good approach to automate these experiments is to compare a fault-free or golden simulation versus as many as faulty ones the designer would consider necessary. The faulty simulation flow usually consists of three main phases [15] [16]: pre-FI, FI and post-FI. The main framework blocks are illustrated in Fig. 2.

- *Pre-Fault Injection Phase*
  - Fault list:this library is created based on a framework environment: Fault Model, Fault Location or Signal Target, FI Time and Fault Duration. Parsing of the de- sign can be applied addressing two different objectives: introduction of faults in random signals or verification that the signal chosen as potential target after conduct- ing the safety analysis, exists. This approach should be able to model any type of the aforementioned faults i.e. permanent, intermittent and transient.
  - Scenario library: The user can select the vehicle scenario to test the correspondent fault list. Speed and initial position of the vehicle can be specified.
  - Fault injector (F): This module injects the previously selected fault from the fault list and the user selected scenario into the vehicle model. Depending on the selected solution, simulator commands or saboteurs techniques might be applied.
- *Fault Injection Phase*
  - This process is controlled by the Fault injector. System Behavioral Models are run following golden-versus- fault simulations approach.
  - Data collector: This functionality performs data collec- tion by means of specific readouts.
- *Post-Fault Injection Phase*
  - Result analyzer: Compare and analyze the recollected data of the faulty target to determine fault effects and FTTIs. After this analysis, the fault tolerance level that the system requires can be balanced together with the coverage of the safety mechanisms.

B. Framework in terms of ISO 26262

Regarding safety assessment in the context of ISO 26262, the aforementioned approach can be used to address the following objectives:

- Safety verification and validation: support or completely proceed with the safety verification and validation of the technical solutions at the different levels of Model, SW, HW-in-the-Loop and vehicle tests by accomplishing an early verification and validation of safety concepts.
- Hazard Analysis and Risk Assessment (HARA): by reproducing a specific driving scenario it is possible to verify that the study done at analysis level is correct and complete. Furthermore, as it is not always an easy task to determine the controllability value of a specific traffic situation, simulating driving scenarios helps the safety engineer to determine the controllability in a more precise way. This is especially relevant in highly automated driving, based on the automation level of the Standard SAE J3016.
- Safety Analysis: as previously pointed out, analytical results are sometimes not sufficient and techniques such as Failure modes and Effect Analysis (FMEA) must be either verified or completed by FI tests.
- FTTI: calculated by measuring the time frame between the inserted fault and the loss of controllability (delay between the fault activation and the violation of a safety goal). If the time constraints derived from the FTTI experiments are so tight, then redundancy might be needed for those components/systems considered as potential fault source. As consequence, a possible safe state, wrapped up by testing data, can be derived.

IV. Use Case: Lateral Control

A. Automated vehicle control architecture

Nowadays, the algorithms embedded in automated driving applications are marked by the integration of different subsystems on a modular architecture. This separation in different modules reduces the time of troubleshooting possible failures. The architecture used in the framework of the current contribution is shown in Fig. 3 and it has the six common modules of automated vehicle control architecture[17]. Those are acquisition, perception, communication, decision, control and actuators. Cooperative maneuvers are not considered in the work, so communication block will be considered in future applications:

noitemsep

- *Acquisition:* It gathers the information of the different sensors in the vehicle. Position and speed, among others, are obtained for example from sensors like DGPS,

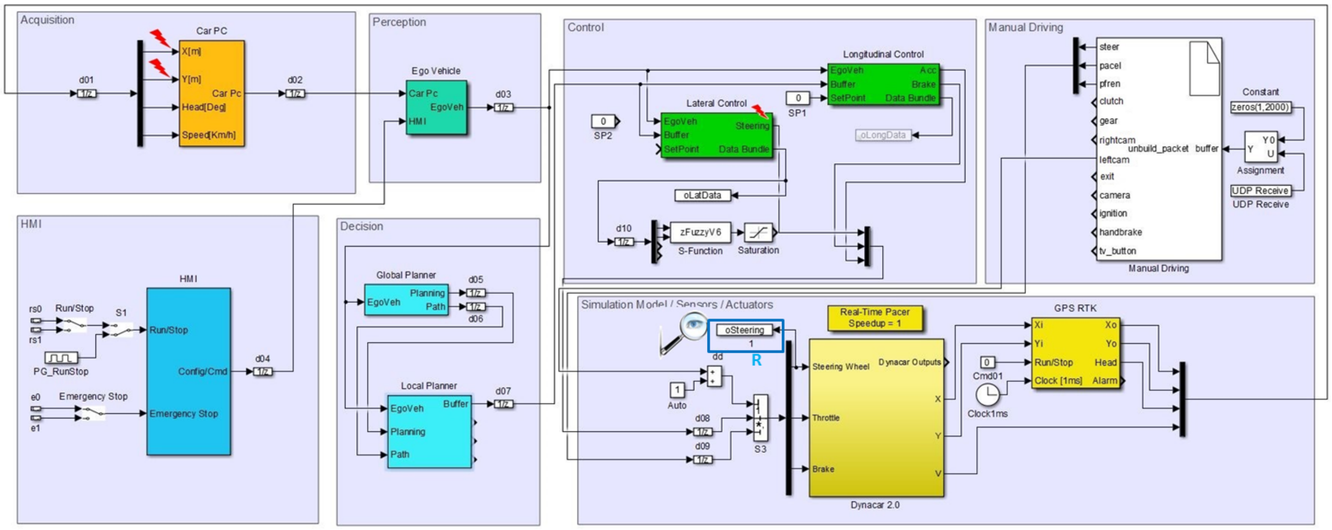


Fig. 3. Control Architecture for automated Vehicles

odometer, Inertial Measurement Units (IMU), lasers on the real vehicle or in the simulation environment. For the purposes of the current approach the information of the Differential GPS + IMU is relevant, because it is related with the calculation of the lateral error.

- *Perception:* The perception module gathers the information from the acquisition module, to process this data and to give reliable information of the positioning and obstacle around.
- *Decision:* This module generates the trajectories that are tracked by the vehicle as in [18], and [19]. Based on this information and the GPS position is calculated the lateral error that will be used in the lateral control law.
- *Control:* The control module receives the data from the decision and it processes this information to send the steering, acceleration and braking assignments to the low level on the vehicle (actuators) or simulation. noitemsep
  - Longitudinal control: it controls the acceleration and braking assignments in the vehicle.
  - Lateral Control: it controls the steering wheel using as reference the lateral error. Different control laws have been used in previous works [20] and [18]. The control law used in this work includes the lateral error (elat), but also the angular error (eang)and the curvature, as in [19]:

$$C_v = K_1 * e_{lat} + K_2 * e_{ang} + K_3 * Curvature \quad (1)$$

  This part of the architecture is relevant for the purposes of the current work. It receives the injection of faulty signal on the steering reference and the introduction of faulty GPS signals to produce wrong calculation of the lateral error that will have a reaction on the steering. This is made with the main goal of producing an evaluation of the function robustness in terms of the controllability.

- *Simulation model, Sensors, Actuators*: In the current approach, the tests have been made using a dynamic model simulation platform of the vehicle (Dynacar) to test the architecture and how it responds against failures.
- *Manual driving:* The architecture considers, additionally, the interaction with a human driver in the control loop.

B. Pre-Fault Injection Phase

This phase covers the tasks of completing the fault list. To do so, as depicted in Figure 6, a preliminary safety analysis (at concept and system level) has been used as starting point. In this way, the possible fault list is collapsed and only potential faults are taken under consideration. Of course, this is extensible to some other component failures and not only to the DGPS + IMU input. Regarding the operational situation, a driving situation where the vehicle is driving at 45km/h maximum in a city with fluent traffic and performing the steering maneuver in a curve at a city intersection is assumed as the most relevant scenario. ISO 26262 specifies how the safety goals are determined against identified hazards and their ASIL (Automotive Safety Integrity Level) derived according to the determinations along 3 dimensions: Exposure (E0 to E4), Severity (S0 to S3) and Controllability (C0 to C3). In this case, the Severity (S) is considered as S2. The reason behind is that as it is more than 10

The PASS/FAIL criteria of the simulation results is defined as the following safety goal "avoid unwanted lateral control when Lateral Error is LateralErrormax". Lane deviation criteria is calculated in the following way (see also Fig. 4):

$$LateralError_{max} =$$
$$(Lane_{Width} - Vehicle_{Width})/2 = (2,5-1.19)/2 = 0,655m$$

C. Fault Injection Phase

The selected failure modes are toggled by introducing extra behavioral models (saboteurs) reproducing those faults

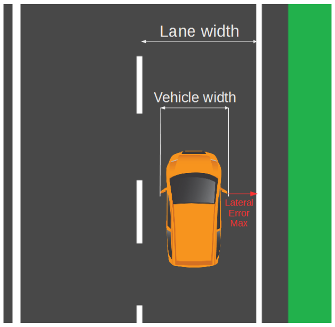


Fig. 4. Lane Derivation Criteria

at the appropriate injection time.They effectively represent different failure modes of the DGPS and lateral control. By applying the process explained in Section III, a golden simulation for each of the selected experiments has been created and different faulty ones representing the previous circumstances. It is worth noting that even if Simulink has been chosen, this approach can be implemented on some other languages as SCADE. Fig. 5 illustrates golden and faulty values for X and Y (DGPS).

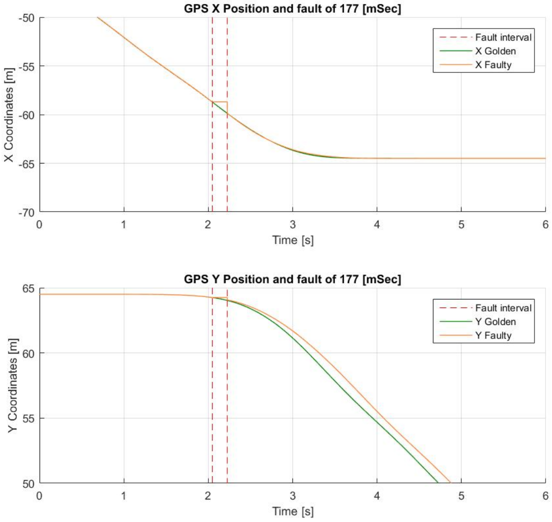
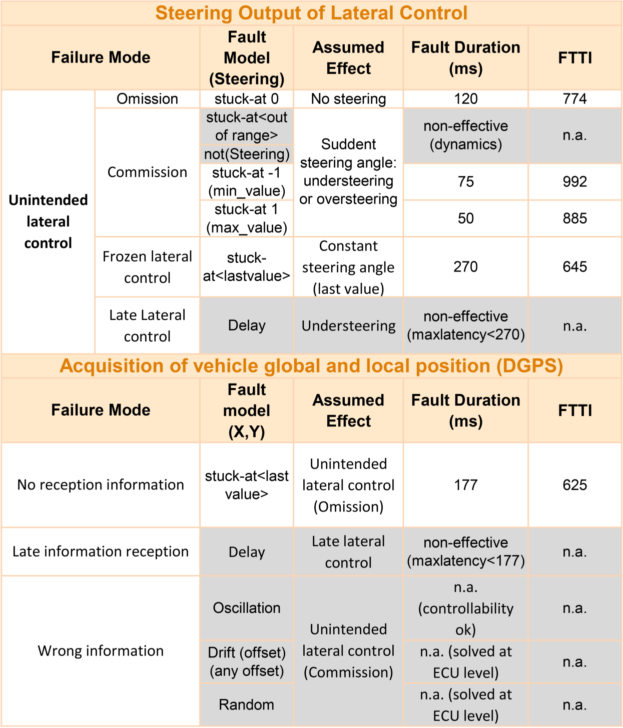


Fig. 5. Simulation results, target signal X,Y

D. Post-Fault Injection and Results

The result analyzer evaluates the collected data based on the PASS/FAIL criteria of each set of experiments. To do so, the results of the so-called golden simulation (without any injected faults) are compared versus faulty ones based on the set read-outs of the experiments, i.e., measurable vehicle dynamic parameters (yaw rate change, derivation from lane center change or lateral accelerations. Fig. 6 depicts the safety analysis and the collection of the results for faults introduced in X,Y DGPS signals and in the steering (see Fig. 3).



Steering Output of Lateral Control

Failure Mode		Fault Model (Steering)	Assumed Effect	Fault Duration (ms)	FTTI
Unintended lateral control	Omission	stuck-at 0	No steering	120	774
	Commission	stuck-at<out of range>	Suddent steering angle: understeering or oversteering	non-effective (dynamics)	n.a.
		not(Steering)			
		stuck-at -1 (min_value)		75	992
		stuck-at 1 (max_value)		50	885
	Frozen lateral control	stuck-at<lastvalue>	Constant steering angle (last value)	270	645
	Late Lateral control	Delay	Understeering	non-effective (maxlatency<270)	n.a.

Acquisition of vehicle global and local position (DGPS)

Failure Mode	Fault model (X,Y)	Assumed Effect	Fault Duration (ms)	FTTI
No reception information	stuck-at<last value>	Unintended lateral control (Omission)	177	625
Late information reception	Delay	Late lateral control	non-effective (maxlatency<177)	n.a.
Wrong information	Oscillation	Unintended lateral control (Commission)	n.a. (controllability ok)	n.a.
	Drift (offset) (any offset)		n.a. (solved at ECU level)	n.a.
	Random		n.a. (solved at ECU level)	n.a.

Fig. 6. FI simulation results: failure effects and FTTI (ms)

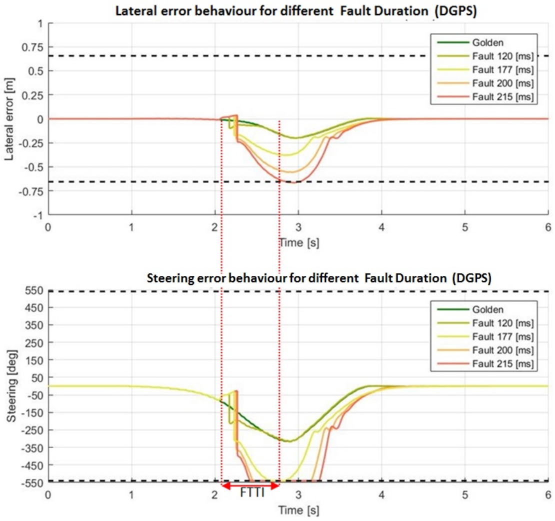


Fig. 7. Results for a faulty DGPS

Fig. 7 and Fig. 8 illustrate how FTTI values have been obtained. To get those values, different fault durations are tested and these double criteria checked: lateral error shall not exceed 0,655 m value and the steering shall not be saturated.

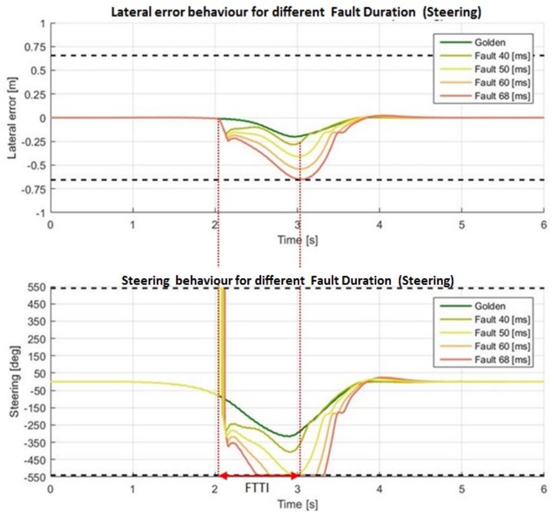


Fig. 8. Results for a faulty steering

In the evaluation of safety concepts, we can consider the worst case scenarios in Steering and acquisition outputs (FTTI 645ms and 625 respectively, see Fig. 6), since FTTI is derived as the stringent time response for each scenario. Both values are similar (less than 2 cycles of the low level frequency, fixed at 10 ms) due to the processing time of the Decision and control block, when failures on DGPS are detected. On the other hand, the processing time of more complex dynamic conditions, for instance when the stuck at the steering at -1 and 1, even if the maximum lateral angular speed is limited. In this case, the resulting value is higher than in the other cases because of the dynamics of the wheels. One of the main outcomes is that the lateral control system can tolerate a permanent DGPS failure for 177 ms without losing vehicle controllability. In the same way, the assumed potential failure effects have been verified for the Lateral Control.

V. Conclusion and Future Work

We have presented a simulation-based FI approach for safety assessment of automated vehicle functions. Our approach has been evaluated on a use case for the model-based design of a lateral control function embedded in an urban vehicle. From a novelty standpoint, we focused on the determination of the fault detection interval for permanent faults based on the maximum lateral error and steering saturation, as a vehicle controllability property. A major strength of the method introduced in this paper is its integration with HARA activities, which enables a seamless ISO 26262-compliant safety assessment process. Our future work spans the spectrum from relaxing the fault simulation constraints to instrumenting the automated assessment work. This includes: (1) to add the capability of collapsing and automating the injection of faults at post-processing stage, (2) the definition of generic fault models to be ready available in a database, (3) the evaluation of the acceptable time for switching the control to the driver while keeping controllability, and (4) to increase the automation of the full fault injection process from HARA to the generation of assessment reports.

Acknowledgment

Authors wants to thank to the H2020 UnCoVerCPS Project (with grant number 643921) and the ECSEL JU AMASS project under H2020 grant agreement No 692474 and from MINETUR (Spain).

References

[1] A. Ruiz, G. Juez, P. Schleiss, and G. Weiss, "A safe generic adaptation mechanism for smart cars," in *Software Reliability Engineering (ISSRE), 2015 IEEE 26th International Symposium on*, pp. 161–171, Nov 2015.

[2] A. Benso and P. Prinetto, *Fault injection techniques and tools for embedded systems reliability evaluation*. Frontiers in electronic testing, Boston, Dordrecht, London: Kluwer academic publ. cop., 2003.

[3] A. Benso and D. C. S., "The art of fault injection," *Journal of Control Engineering and Applied Informatics*, pp. 9–18, 2011.

[4] R. Svenningsson, "Model-implemented fault injection for robustness assessment," 2011.

[5] J. Vinter, L. Bromander, P. Raistrick, and H. Edler, "Fiscade - a fault injection tool for scade models," in *Automotive Electronics, 2007 3rd Institution of Engineering and Technology Conference on*, pp. 1–9, June 2007.

[6] D. Johansson and P. Karlsson, "Safety mechanisms for random ecu hardware failures in compliance with iso 26262," *Master of Science Thesis in Embedded Electronic System Design*, 2015.

[7] A. Silveira, R. Araujo, and R. De Castro, "Fieev: A Co-Simulation Framework for Fault Injection in Electrical Vehicles," in *2012 Ieee International Conference on Vehicular Electronics and Safety, Icves 2012*, pp. 357–362, 2012. Citations: crossref, scopus.

[8] P. K. . M. Wagner, "Challenges in autonomous vehicle testing and validation," *2016 SAE World Congress*, 2016.

[9] J. Arlat, A. Costes, Y. Crouzet, J.-C. Laprie, and D. Powell, "Fault injection and dependability evaluation of fault-tolerant systems," *IEEE Transactions on computers*, vol. 42, 1993.

[10] J.-C. Laprie, "Dependable computing: Concepts, limits,challenges," *IEEE International Symposium on Fault-Tolerant Computing*, pp. 42 – 54, 1995.

[11] R. A. Haissam Ziade and R. Velazco, "A survey on fault injection techniques," *The International Arab Journal of Information Technology*, 2004.

[12] M. Pintard, "Des analyses de securite a la validation experimentale par injection de fautes - le cas des systemes embarques automobiles," *Institut National Polytechnique de Toulouse*, 2015.

[13] I. Iglesias, "Herramienta para acelerar el desarrollo de nuevos sistemas y controles para automocin," *Revista Automtica e Intrumentacin*, vol. 478, pp. 45 – 49, 2015.

[14] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins, and D. Powell, "Fault injection for dependability validation: A methodology and some applications," *IEEE Trans. Softw. Eng.*, vol. 16, pp. 166–182, Feb. 1990.

[15] C. Novello and G. J. Uriagereka, "Practical implementation of a fault injection methodology using a simulation based approach in the framework of iso 26262," *edaWorkshop 14*, 2014.

[16] S. D. C. A. Benso, A. Bosio and R. Mariani, "A functional verification based fault injection eviroment," *22nd IEEE International Symposium on Defect and Tolerance in VLSI Systems*, 2007.

[17] D. González and J. Pérez, "Control architecture for cybernetic transportation systems in urban environments," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1119 – 1124, 2013.

[18] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 1135 – 1145, 2015.

[19] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 510 – 515, 2014.

[20] J. Pérez, F. Nashashibi, B. Lefaudeux, P. Resende, and E. Pollard, "Autonomous docking based on infrared system for electric vehicle charging in urban areas," *Sensors journal*, pp. 2645 – 2663, 2013.